Attorney's Docket No.: 10559-580001
Intel Docket No.: P11134

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

#6

Applicants : Adam T. Lake, et al.
Serial No. : 09/982,475
Filed : October 17, 2001/
Title : GENERATING A SHADOW FOR A THREE-DIMENSIONAL MODEL

Art Unit : 2676
Examiner : Gregory F. Cunningham
Assignee : Intel Corporation

Mail Stop Non-Fee Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION OF ADAM T. LAKE AND CARL S. MARSHALL UNDER 37 C.F.R § 1.131

1. We are co-inventors of the above-identified application and are employed by the current assignee of the application, Intel Corporation.

2. Attached is a copy of an invention disclosure form (with non-descriptive portions redacted) that describes the invention of the subject application, and which is date-stamped December 2000. This date is prior to the effective date of U.S. Application Publication No. 2002/0061194 (Wu), namely July 3, 2001. .

3. Accordingly, we hereby declare that the invention was conceived of, and reduced to practice, prior to the effective date of Wu.

4. We also hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that those statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment or both under section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Adam T. Lake

9/3/03
Date

Carl S. Marshall
Carl S. Marshall

9-3-03
Date

P11134

17211

10559-580001

INTEL INVENTION DISCLOSURE

DATE: __12/12/2000__

Software/
Internet / IAL/ MAL

It is important to provide accurate and detailed information on this form. The information will be used to evaluate your invention for possible filing as a patent application. When completed, please return this form to the **Legal Department at JF3-147**. If you have any questions, please call **264-0444 or 264-1476.**

1. Inventor: ________Lake____________________ ___________Adam______________________ ______T______
Last Name First Name Middle Initial

Phone ____503-264-8299__________ M/S: __JF278__________ Fax # ____264-9372________________

Citizenship: ______USA______________ WWID ___10599683_______

Home Address: ___15451 NW Andalusian Way.___ City ____Portland________ State _OR_ Zip __97229_______

Group: (e.g. TMG, NBG, CEG) ____IAL__________ Division Name ____MAL___________ Subdivision _____G3D____

Supervisor*__Sanjeev Trika______________ WWID ___10507137_______ Phone _503 264 7486__ M/S: ___JF2-78____

Inventor: _____Marshall____________________ __________Carl______________________ ________S_____
Last Name First Name Middle Initial

Phone ____503-712-1245__________ M/S: ____JF2-78__________ Fax # ____264-9372______________

Citizenship: _____USA______________ WWID __10503453________

Home Address: ___15451 NW Andalusian Way________ City ___Portland__________ State _OR_ Zip ___97229______

Group: (e.g. TMG, NBG, CEG) ____IAL__________ Division Name _____MAL___________ Subdivision ___G3D______

Supervisor*__Sanjeev Trika______________ WWID ___10507137_______ Phone _503 264 7486__ M/S: ___JF2-78____

(PROVIDE SAME INFORMATION AS ABOVE FOR EACH ADDITIONAL INVENTOR)

2. **Title of Invention:** Bones Based Shadows in Real-Time Character Animation Systems

7. Is this invention related to any other invention disclosure that you have recently submitted? If so, please give the title and inventors: No.

PLEASE READ AND FOLLOW THE DIRECTIONS ON HOW TO WRITE A DESCRIPTION OF YOUR INVENTION

Please attach a page to this form, DATED AND SIGNED BY AT LEAST ONE PERSON WHO IS NOT A NAMED INVENTOR, to provide a description of the invention, and include the following information:

1. **Describe in detail what the components of the invention are and how the invention works.**
2. **Describe advantage(s) of your invention over what is done now.**
3. **YOU MUST include at least one figure illustrating the invention. If the invention relates to software, include a flowchart or pseudo-code representation of the algorithm.**
4. **Value of your invention to Intel (how will it be used?).**
5. **Identify the closest or most pertinent prior art that you are aware of.**
6. **Who is likely to want to use this invention or infringe the patent if one is obtained and how would infringement be detected?**

***HAVE YOUR SUPERVISOR READ, DATE AND SIGN COMPLETED FORM**

DATE: 12.13.00 SUPERVISOR: Pamela Luscarch

BY THIS SIGNING, I (SUPERVISOR) ACKNOWLEDGE THAT I HAVE READ AND UNDERSTAND THIS DISCLOSURE, AND RECOMMEND THAT THE HONORARIUM BE PAID

Bones Based Shadows in Real-Time Character Animation Systems

Adam Lake and Carl Marshall

The current generation of game engines are a leap forward in photorealism. They support a vast array of techniques for generating these realistic images. Each step closer we take towards realizing an image indistinguishable from a photograph, we increase the computational workload we place on the CPU. In addition, we increase the complexity of our codebase with new datastructures, new conditions, and new bugs to track down. In many ways, the graphics engineers problem is an optimization exercise: obtain the most photorealistic image possible while minimizing new conditions and bugs into the project.

1.0 Description of Invention

Our new technique creates photorealistic shadows of scene geometry efficiently and in real-time without introducing new data structures into the codebase. In most 3d game engines currently in use or in development, the characters have an underlying bones based representation that is used for animation. The technique takes advantage of this *already existing* data structure to cast shadows on scene geometry. Additionally, it is an advantage over previous shadow generation algorithms because it *minimizes the amount of computation* that needs to take place to compute the shadow. By adjusting the input to the shadow creation phase of the algorithm, there is a great deal of control that can be put in place by the graphics engineer when deciding the artistic look and feel of the shadow. The result is a shadow that looks as realistic as the current state of the art while *minimizing computation and maximizing developer control* over the look and feel of the shadow.

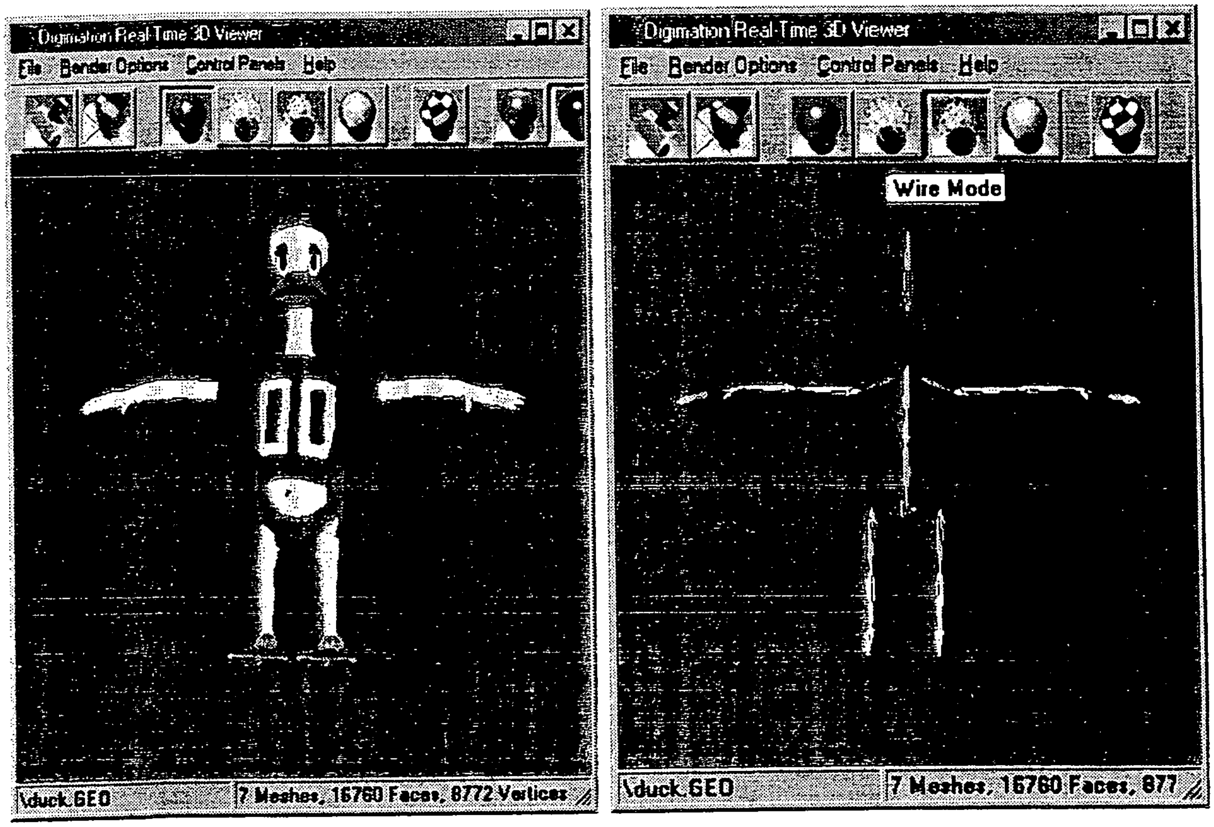
Digimation Real-Time 3D Viewer
File Render Options Control Panels Help
\duck.GEO
7 Meshes, 16760 Faces, 8772 Vertices
Digimation Real-Time 3D Viewer
File Render Options Control Panels Help
Wire Mode
\duck.GEO
7 Meshes, 16760 Faces, 877

Figure 1. On the right, an image of a character from a 3D animation. On the right, the underlying bones that are used to animate the character. Since the bones have to be read in to do keyframe animation already, we re-use the bones to approximate realistic looking shadows that closely represent the geometry of the character while minimizing computation and complexity of the codebase.

For the purposes of this disclosure, we will break the 3d game engine down into the load time and run time components. Note that this is a very general but sufficient description of the rendering system.

At load time:

1. Read in geometry (3d information for all meshes used in the scene).
2. Read in animation (bones and animation data).

At run-time:

1. Determine location of camera.
2. Transform geometry, performing all necessary animation timesteps for the bones
3. Render Geometry taking into account positions of lights, characters, objects, and shadows in the scene.

The algorithm for shadow generation can be written as follows:

Bones Based Shadow Generation Algorithm

At load time:

1. Request references to bones from animation engine. Later, this information will be used to obtain transformed positions of bones. Used in steps 1-4 of the run-time.
2. Compute bounding volume of bone, used in step 5 of the run-time.

Run-Time:

1. Request from the animation engine updated position of bones given that animation in step 2 of the run-time has taken place.
2. Taking into account the positions of the lights and camera in the scene, project each bone onto the geometry of the scene as if it was projected from the light to the ground. This is done by simply drawing a line for each endpoint of the bone from the light to the surface it intersects, as shown in Figure 2. These two endpoints serve as the points from which we 'grow' a new piece of geometry that we will later *texture map* with a shadow image..
3. Using the endpoints of the bone as a medial axis, project outward from the bone to create a quadrilateral[1] in the same plane as the geometry that we intersect. The size of the polygon we grow is dependent on a user specified weighting and the length of the bone.
4. Color the polygon the desired color. Transparency should also be set so that the color of the underlying geometry shines through correctly. Usually this means an alpha (transparency) value of around 50%. Additionally, we will use the quadrilateral to texture map any desired shadow. For example, we could have a shadow by projecting an intensity map that is fuzzy around the edges. This is called *soft shadowing* and is normally prohibitively complex to perform in real-time computer graphics engines. Using our technique it is made feasible in real-time on current computer hardware. An example of a soft shadow is presented in Figure 3.
5. To determine the size of the projected shadow, we can use several techniques. One technique is to determine an average bounding volume of the bone at load time and use these bounds when we project the shadow. An additional advantage of this technique is that no user intervention is required. Another technique would allow the user to adjust the amount of shadow after projection. This technique would also take into account the distance from the occluding geometry to the target surface. Yet another method would be some combination of both of these methods.

[1] We will make the assumption in this disclosure that we are using a quadrilateral. It should be pointed out that this is NOT necessary for the algorithm. Other types of geometry would also be useful as discussed on page 3.

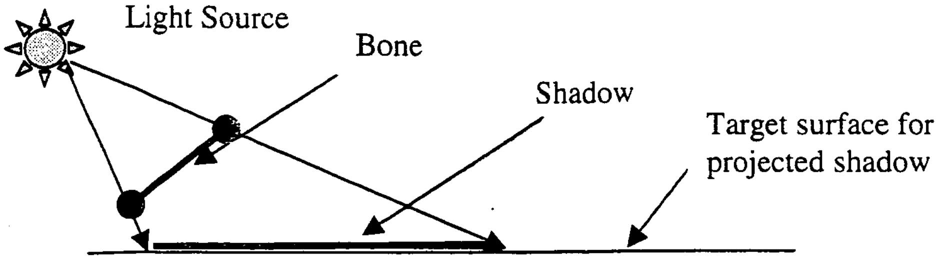

Figure 2. This figure demonstrates in 2D steps 1-2 of the run-time process. The bone is used to project an image onto the target surface to determine the position and orientation of the shadow.

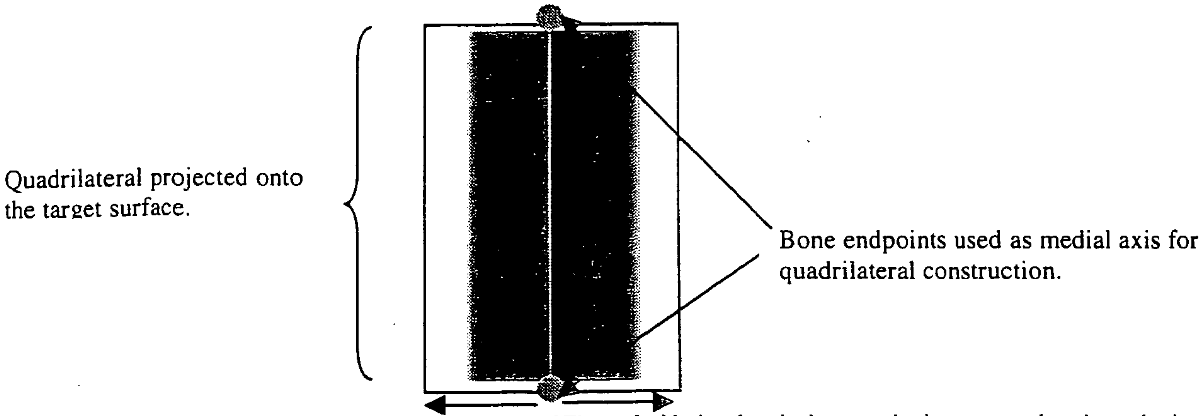
Quadrilateral projected onto
the target surface.
Bone endpoints used as medial axis for
quadrilateral construction.

Figure 3. Quadrilateral projected onto the target surface of Figure 2. Notice that the bone endpoints are used as the endpoints of a quadrilateral projected onto the target surface. The quadrilateral extends outward from the medial axis in the plane of the target surface and the dark texture (shadow) is applied to the quadrilateral. Notice that we can tailor the type of texture that we can apply. In this case the intensity of the shadow tapers off near the edges creating a soft shadow, computationally prohibitive in current 3d game architectures.

A fundamental assumption of the algorithm is that this is used in combination with character animation. Therefore, we can assume that the bone is very similar to that in a human or other biological or biologically inspired creature. This is sometimes important because certain types of artistic hacks will fail. For example, an artist might animate an object like a ball with a bone down the middle. The shadow of the ball might look odd because the projected shadow is more cylindrical with sharp edges.

A fix for this assumption is to allow the artist or programmer to have different types of projections to choose from. As previously mentioned, they already will have the ability to adjust the width, temperature, and color of the shadow projected. This feature would add the ability to choose to project a cylindrical, elliptical, or other geometric shadow projections.

2.0 Describe advantage(s) of your invention over what is done now.

Our technique has two key advantages. First, we make use of already existing data structures that are already available rather than having to create complicated storage mechanisms. Second, we minimize computation, and are therefore faster. In the usual technique for realistic shadow generation *each polygon* of an object needs projected onto the geometry. Using this method we reduce the number of projections to ONLY the bones. As the density of polygons in 3D engines increases, techniques such as ours are of increasing value.

3.0 YOU MUST include at least one figure illustrating the invention. If the invention relates to software, include a flowchart or pseudo-code representation of the algorithm.

See figures and algorithm in Section 1. In addition, we have another demonstrated use of such an algorithm. In this example, the projection scheme is used to again decrease the amount of geometry that needs rendering for while increasing the amount of realism in the scene. Here, we create a virtual light source about a tree that is used to project a 'shadow' of leaves onto the ground.

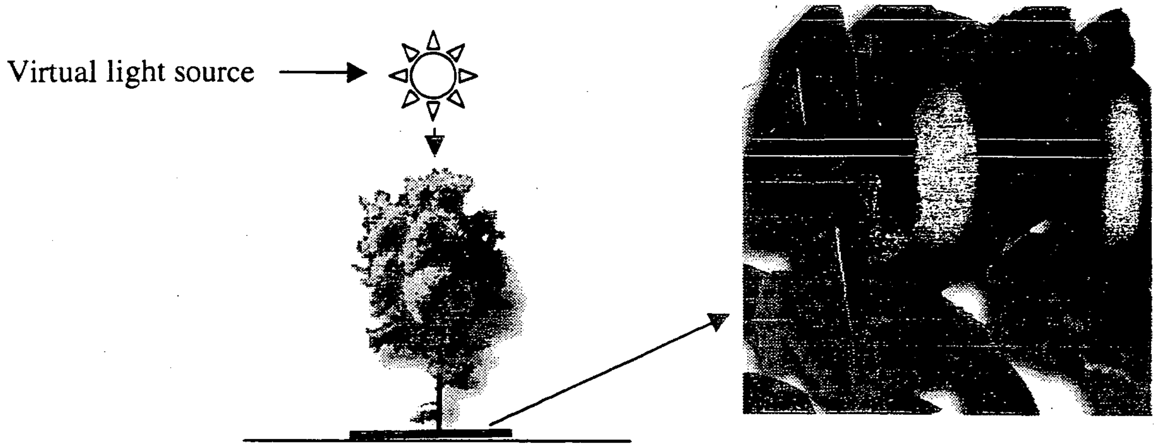

Figure 4. Another use of the projected shadow technique. In this example, a virtual light source is placed above a tree which has a virtual shadow of 'leaves' projected to the target surface. Normally, leaves that look like this would be prohibitively complex due to the large number of polygons it would take to render them. In a virtual world, the light source can be used as a source that 'rains' leaves down from the tree over time. As time moves on, more and more leaves would fall. Also, the colors of the leaves could change as they are rendered into the texture map that is projected from the virtual light source. In addition, for our algorithm to be used, the tree trunk and some of the larger branches can be used as the bones based representation of the geometry for projection!